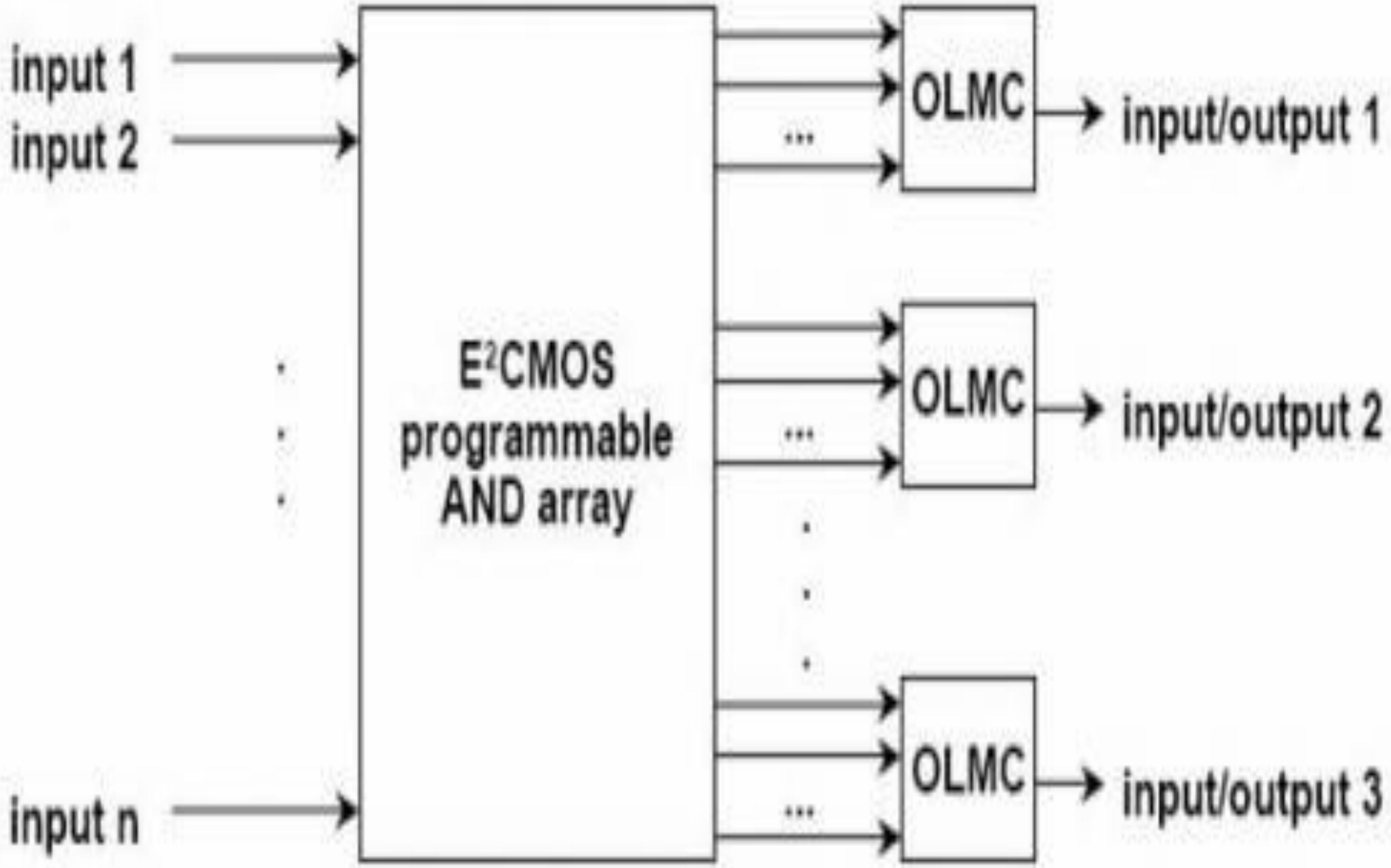
Generic Array Logic (GAL):

- Generic array logic family consists of electrically erasable programmable devices designed by lattice semiconductor.
- Same logic properties as PAL but can be erased and reprogrammed.
- Programmed and reprogrammed using a PAL programmer
- It has a fixed OR array and a programmable And array the reprogrammable array is essentially a grid of conductors forming rows and columns with an electrically erasable CMOS (E2CMOS) cell at each cross point.
- The GAL has the programmable logic and the OLMC (Output Logic Macro cell) Logic that excludes OR gates and flip-flops.


input 1
input 2
input n
E²CMOS programmable AND array
OLMC
OLMC
OLMC
input/output 1
input/output 2
input/output 3

Generic Array Logic – Macrocell

- I/O circuit that can be configured as a registered output, a combinational output, or a dedicated input as required.
- Outputs can also be specified as active-HIGH or active-LOW.

PEEL (Programmable Electrically Erasable Logic) Device :

- Programmable Electrically Erasable Logic
- Introduced by the international CMOS technology (ICT) corporation.
- ICT offer the most flexible PLD solutions for lower pin count application.
- Include PEEL devices, PEEL array and PEEL development tool.
- PEEL devices are another family of devices that are intended as PAL replacements the PEEL is available in 20 pin different packages with speeds ranging from 5ns to 25ns.
- The PEEL architecture allows it to replace over 20 standard 20 pin PLDs (PAL, GAL, etc.)

- Features of PEEL:
- 1. Speed ranging from 5ns to 25ns
- 2. Low Power consumption.
- 3. CMOS Electrically Erasable Technology.
- 4. Reduces development Cost
- 5. Flexible architecture.

Design implementation using CPLD and FPGAs

- Implementing a logic design with the FPGA or CPLD development software usually consists of the following steps :

1. You enter a description of your logic circuit using a *hardware description language* (HDL) such as VHDL or Verilog. You can also draw your design using a schematic editor.
2. You use a *logic synthesizer* program to transform the HDL or schematic into a *netlist*. The netlist is just a description of the various logic gates in your design and how they are interconnected.

Cont..

3. You use the *implementation tools* to map the logic gates and interconnections into the FPGA. The configurable logic blocks in the FPGA can be further decomposed into *look-up tables* that perform logic operations. The CLBs and LUTs are interwoven with various *routing resources*. The mapping tool collects your netlist gates into groups that fit into the LUTs and then the *place & route tool* assigns the gate collections to specific CLBs while opening or closing the switches in the routing matrices to connect the gates together.
4. Once the implementation phase is complete, a program extracts the state of the switches in the routing matrices and generates a *bitstream* where the ones and zeroes correspond to open or closed switches.

Cont..

5. The bitstream is *downloaded* into a physical FPGA chip (usually embedded in some larger system such as an XS Board). The electronic switches in the FPGA open or close in response to the binary bits in the bitstream. Upon completion of the downloading, the FPGA will perform the operations specified by your HDL code or schematic. You can apply input signals to the I/O pins of the FPGA to check the operation of your design.